

INTERRUPT-PROCESSING SYSTEM FOR SHORTENING INTERRUPT LATENCY IN MICROPROCESSOR

BACKGROUND OF THE INVENTION

5 1. Field of the invention

The present invention relates to a data processing system, and more particularly, the present invention relates to a data processing system for the interrupt latency of the microprocessor.

2. Description of the prior art

10 In the microprocessor used in common electronic, non-personal computer, devices such as a digital camera, the interrupt vector table of the data processing system comprises a set of vector addresses; each address stores an entry instruction of the interrupt service routine (The entry instruction is the first instruction of the whole interrupt service routine). When the central processing unit (CPU) accepts an
15 interrupt request, the CPU reads the interrupt vector table and executes the interrupt service routine corresponded to the interrupt request.

In this type of data processing system, because the address of the reset vector is next to the other vectors, when booting the microprocessor, the CPU fetches the first program instruction of the whole application program in the reset vector to execute;
20 therefore, the reset vector must be located in the non-volatile memory, such as the erasable programmable read-only memory (EPROM) or the flash read-only memory (Flash ROM)...etc.

However, when the CPU reads the non-volatile memory, the reading speed is

slow; therefore, in real practice, the whole application program, comprised of the interrupt service routines, will be copied into the higher speed volatile memory to be executed by the CPU for increasing the executing efficiency.

Referring to FIG.1, FIG. 1 is a function block diagram of a conventional data processing system 1. The data processing system 1 comprises a set of memory modules 30 for storing program instructions and data, a microprocessor 20, a power source 28, and a bus 60.

The memory modules 30 comprise a low-speed memory 32 and a high-speed memory 34. The low-speed memory 32 stores an interrupt vector table 36 for recording at least one entry instruction of an interrupt service routine.

The power source 28 is used for providing electrical power to the data processing system 1. The power source 28 comprises a switch 29. When the power source 28 is shut down, the program instructions and data stored in the high-speed memory 34 are lost; however, the program instructions and data stored in the low-speed memory are preserved.

The microprocessor 20 comprises a CPU 22 for executing program instructions and calculating data. The CPU 22 is designed to fetch program instructions in the low-speed memory 32 when an interruption occurs.

The bus 60 is used for connecting to the CPU 22 and the memory modules 30 for transmitting program instructions and data.

When an interruption occurs, the CPU 22 fetches the corresponding entry instruction of the interrupt service routine in the interrupt vector table 36 of the low-speed memory 32.

By the restriction of the basic architecture of the CPU, when an interruption occurs, the CPU fetches the entry instruction of the interrupt service routine in the

low-speed non-volatile memory, and then branches to the rest of program located in the high speed memory. Although this method is commonly used, the interrupt service efficiency of the CPU is restricted. If the interrupt latency can be reduced, the system executing efficiency can be improved.

5

SUMMARY OF THE INVENTION

The objective of the present invention is to provide a data processing system for reducing the interrupt latency, so as to increase the interrupt service efficiency.

According to an embodiment of the present invention, the data processing system comprises a set of memory modules for storing program instructions and data, a microprocessor, a power source for providing electrical power to the data processing system, and a bus.

The set of memory modules comprise a low-speed memory and a high-speed memory. Both the low-speed memory and the high-speed memory store an interrupt vector table 36 individually for recording at least one entry instruction of the interrupt service routine.

The power source comprises a switch. When the power source is shut down, program instructions and data stored in the high-speed memory are lost; however, program instructions and data stored in the low-speed memory are preserved.

The microprocessor comprises a central processing unit (CPU) for executing program instructions and calculating data. The CPU is designed to fetch the program instructions in the low-speed memory when an interruption occurs.

The microprocessor further comprises a memory controller; under the memory controller's control, the CPU fetches the program instruction and accesses the data in the set of memory modules. The memory controller also comprises a re-addressing

device.

The bus connects to the CPU, the memory controller, and the set of memory modules for transmitting program instructions and data.

When the interruption occurs, the CPU generates an interrupt vector address to
5 the memory controller. If the re-addressing device of the memory controller identifies
that the address falls within the address range of the interrupt vector table, the re-
addressing device sends out an enable signal to the high-speed memory which
enables the CPU to fetch the corresponding entry instruction of the interrupt service
routines in the high-speed memory, instead of the predetermined low-speed memory,
10 so as to reduce the interrupt latency when fetching the program instruction.

The data processing system of the present invention, through the re-addressing device, allows the CPU to directly execute the interrupt service routine in the high-speed memory, thus not needing to fetch the entry instruction of the interrupt service routines in the low-speed memory. Therefore, the interrupt latency can be reduced.

15 The advantage and spirit of the invention may be understood by the following recitations together with the appended drawings.

BRIEF DESCRIPTION OF THE APPENDED DRAWINGS

FIG. 1 is a function block diagram of a conventional data processing system.

FIG. 2 is a function block diagram of a data processing system according to the
20 present invention.

FIG. 3 is a function block diagram of the re-addressing device of the data processing system shown in FIG. 2.

FIG. 4 is a function block diagram of another data processing system according

to the present invention.

FIG. 5 is a function block diagram of the re-addressing device of the data processing system shown in FIG. 4.

DETAILED DESCRIPTION OF THE INVENTION

5 Referring to FIG. 2, FIG. 2 is a function block diagram of a data processing system 2 according to the present invention. The data processing system 2 comprises a set of memory modules 30 for storing program instructions and data, a microprocessor 20, a power source 28 for providing electrical power to the data processing system 2, and a bus 60.

10 The set of memory modules 30 comprise a low-speed memory 32 and a high-speed memory 34. Both the low-speed memory 32 and the high-speed memory 34 store an interrupt vector table 36 individually for recording at least one entry instruction of an interrupt service routine 38. In this embodiment, there is one interrupt service routine 38. The entry instruction is the first instruction of the whole
15 interrupt service routine 38.

The power source 28 comprises a switch 29. When the power source 28 is shut down, program instructions and data stored in the high-speed memory 34 are lost; however, program instructions and data stored in the low-speed memory 32 are preserved.

20 The low-speed memory 32 is a non-volatile memory, such as the electrical programmable read-only memory (EPROM) or the flash read-only memory (Flash ROM). The interrupt vector table 36 and the interrupt service routines 38 stored in the low-speed memory 32 would not be lost when shutting down the power source 28.

The high-speed memory 34 is an external dynamic random access memory

(DRAM), which is a volatile memory built outside the microprocessor 20. When booting the microprocessor, the interrupt vector table 36 and the interrupt service routine 38 stored in the low-speed memory 32 are copied into the high-speed memory 34. After shutting down the power source 28, the interrupt vector table 36 and the interrupt service routine 38 stored in the high-speed memory 34 are lost. Besides, the high-speed memory 34 can be a built-in static random access memory (SRAM) in the microprocessor (not shown).

The microprocessor 20 comprises a central processing unit (CPU) 22 for executing program instructions and calculating data. The CPU 22 is designed to fetch program instructions in the low-speed memory 32 when an interruption occurs. The microprocessor 20 can be a non-PC (personal computer) architecture microprocessor, such as the microprocessor for the digital camera.

The microprocessor 20 further comprises a memory controller 24; under the memory controller's control, the CPU 22 fetches the program instruction and accesses the data in the set of memory modules 30. The memory controller 24 also comprises a re-addressing device 26A.

The bus 60 is used to connect the CPU 22, the memory controller 24, and the set of memory modules 30 for transmitting program instructions and data.

When the interruption occurs, the CPU 22 generates an interrupt vector address to the memory controller 24. If the re-addressing device 26A of the memory controller 24 identifies that the address falls within the address range of the interrupt vector table 36, the re-addressing device 36A sends out an enable signal to the high-speed memory 34 which enables the CPU 22 to fetch the corresponding entry instruction of the interrupt service routine in the high-speed memory 34, instead of the predetermined low-speed memory 32, so as to reduce the interrupt latency when fetching the program instruction.

Referring to FIG. 3, FIG. 3 is a function block diagram of the re-addressing device 26A of the data processing system 2 shown in FIG. 2. The re-addressing device 26A comprises a multiplexer 42, a first address decoder 44A, and a second address decoder 46A.

5 The first address decoder 44A is used for decoding an original address 50 generated by the CPU 22 to identify whether the original address 50 falls within the address range of the interrupt vector table 36. The first address decoder 44A will also generate a corresponding identifying signal.

10 The multiplexer 42 is used for selecting between the original address 50 generated by the CPU 22 and a predetermined re-directing address 48 as a valid address according to the identifying signal. The re-directing address 48 is programmable and falls within the address range of the high-speed memory 34, distinguishably from the vector addresses of the low-speed memory 32.

15 The second address decoder 46A is used for decoding the valid address generated by the multiplexer 42 to determine whether the enable signal generated by the memory controller 24 is sent to the high-speed memory 34 or the low-speed memory 32.

20 When the first address decoder 44A determines that the original address 50 generated by the CPU 22 falls within the address range of the interrupt vector table 36, the multiplexer 42 replaces the original address 50 by the re-directing address 48 that fell within the address range of the high-speed memory 34 to be the valid address. The second address decoder 46A then sends a high-speed enable signal to enable the high-speed memory 34. Therefore, the CPU 22 directly fetches the entry instruction of the interrupt service routine 38 in the high-speed memory 34, instead of the
25 predetermined low-speed memory 32.

When the first address decoder 44A determines that the original address 50

generated by the CPU 22 does not fall within the address range of the interrupt vector table 36, the multiplexer 42 selects the original address 50 to be the valid address. The second address decoder 46A then sends an enable signal to the memory where the original address 50 is located according to the valid address.

5 Referring to FIG. 4, FIG. 4 is a function block diagram of another data processing system 4 according to the present invention. The main difference between the data processing system 4 and the data processing system 2 is that the data processing system 4 further comprises an on-chip high-speed memory 40. The re-addressing device 26B of the data processing system 4 is different from the re-addressing device 26A of the data processing system 2. The on-chip high-speed memory 40 also comprises an interrupt vector table 36 for recording at least one entry instruction of the interrupt service routine 38, and the vector address range is the same as the vector address range of the interrupt vector table 36 of the low-speed memory 32. The other elements and functions of the data processing system 4 are similar to those of the data processing system 2 and have described above, and would not be described here again.

When an interruption occurs, the CPU 22 generates an interrupt vector address to the memory controller 24. If the re-addressing device 26B of the memory controller 24 identifies that the interrupt vector address falls within the address range of the interrupt vector table 36, the re-addressing device 26B sends out an enable signal to the on-chip high-speed memory 40 which enables the CPU 22 to fetch the corresponding entry instruction of the interrupt service routines 38 in the high-speed memory 40, instead of the predetermined low-speed memory 32, so as to reduce the interrupt latency when fetching the program instruction.

25 Referring to FIG. 5, FIG. 5 is a function block diagram of the re-addressing device 26B of the data processing system 4 shown in FIG. 4. The re-addressing device 26B comprises a first address decoder 44B, a second address decoder 46B, and a XOR (exclusive or) gate 52.

The first address decoder 44B is used for decoding an original address generated by the CPU 22 to determine whether the original address falls within the address range of the interrupt vector table 36, and to generate a corresponding on-chip enable signal to enable the on-chip high-speed memory 40.

5 The second address decoder 46B is used for decoding the original address generated by the CPU 22 to determine whether the original address falls within the address range of the high-speed memory 34 or the low-speed memory 32; then it will correspondingly generate a high-speed enable signal to enable the high-speed memory 34 or generate a corresponding identifying signal of the low-speed memory.

10 The XOR gate 52 is used for receiving the on-chip enable signal and the identifying signal of the low-speed memory to perform exclusive-or operation and correspondingly to generate a low-speed enable signal to enable the low-speed memory 32.

15 Comparing to the prior art, the data processing system of the present invention, through the re-addressing device, allows the CPU to directly fetch the entry instruction of the interrupt service routines in the on-chip high-speed memory, thus not needing to fetch the entry instruction of the interrupt service routines in the low-speed non-volatile memory. Therefore, the interrupt latency can be reduced.

20 With the example and explanations above, the features and spirits of the invention will be hopefully well described. Those skilled in the art will readily observe that numerous modifications and alterations of the device may be made while retaining the teaching of the invention. Accordingly, the above disclosure should be construed as limited only by the metes and bounds of the appended claims.